Context Free Grammar for Regular Expression Language

```
<program>      :=   <regex> $$ | '^'<regex> $$ | <regex>'$' $$ | '^'<regex>'$' $$
<regex>        :=   <term><term_tail> | ε
<term_tail>    :=   '|'<regex> | ε
<term>         :=   <factor><factor_tail>
<factor_tail>  :=   <factor><factor_tail> | ε
<factor>       :=   <base><base_tail>
<base_tail>    :=   <rep_op><base_tail> | ε
<base>         :=   <character> | <charset> | '('<regex>')'
<character>    :=   <ordchar> | <escchar> | '.'
<charset>      :=   '['<text><text_tail>']' | '[''^'<text><text_tail>']'
<text_tail>    :=   <text><text_tail> | ε
<text>         :=   <character> | <charset>
<rep_op>       :=   * | + | ?
```

LL(1) Parse Table

	ORDCHAR(1)	ESCCHAR(2)	.(3)	^(4)	$(5)	[(6)	](7)	((8)	)(9)	|(10)	*(11)	+(12)	?(13)	$$(14)
PROGRAM	√	√	√	√	√	√		√						√
REGEX	√	√	√		√	√		√						√
TERM_TAIL					√				√	√				√
TERM	√	√	√			√		√						
FACTOR_TAIL	√	√	√		√	√		√	√	√				√
FACTOR	√	√	√			√		√						
BASE_TAIL	√	√	√		√	√		√	√	√	√	√	√	√
BASE	√	√	√			√		√						
CHARACTER	√	√	√											
CHARSET						√								
TEXT_TAIL	√	√	√			√	√							
TEXT	√	√	√			√								
REP_OP											√	√	√	